﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace NhanSu

{

public partial class formCongViec : Form

{

SqlConnection connection;

SqlCommand command;

string str = "Data Source=TTTT\\TTTT;Initial Catalog=NhanSu;Integrated Security=True";

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

void loaddata()

{

command = connection.CreateCommand();

command.CommandText = "select \* from CongViec ";

adapter.SelectCommand = command;

table.Clear();

adapter.Fill(table);

dgv.DataSource = table;

}

public formCongViec()

{

InitializeComponent();

}

private void formCongViec\_Load(object sender, EventArgs e)

{

connection = new SqlConnection(str);

connection.Open();

loaddata();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

// Lấy dữ liệu từ các ô trong dòng đã chọn

string maCongViec = dgv.Rows[e.RowIndex].Cells["MaCongViec"].Value.ToString();

string tenCongViec = dgv.Rows[e.RowIndex].Cells["TenCongViec"].Value.ToString();

string moTa = dgv.Rows[e.RowIndex].Cells["MoTa"].Value.ToString();

int uuTien = Convert.ToInt32(dgv.Rows[e.RowIndex].Cells["ƯuTien"].Value);

int trangThai = Convert.ToInt32(dgv.Rows[e.RowIndex].Cells["TrangThaiCV"].Value);

DateTime ngayBatDau = Convert.ToDateTime(dgv.Rows[e.RowIndex].Cells["NgayBatDau"].Value);

DateTime ngayKetThuc = Convert.ToDateTime(dgv.Rows[e.RowIndex].Cells["NgayKetThuc"].Value);

// Chuyển đổi giá trị UuTien từ số sang văn bản

string uuTienText = ConvertUuTienNumberToText(uuTien);

// Gán giá trị vào các điều khiển trong form

txtMaCongViec.Text = maCongViec;

txtTenCongViec.Text = tenCongViec;

tb\_MoTa.Text = moTa;

CbUuTien.Text = uuTienText; // Gán giá trị "Nghiêm trọng", "Cao", "Trung bình", "Thấp" vào ComboBox

rbtHoanThanh.Checked = trangThai == 1; // Kiểm tra trạng thái (1 là Hoàn thành)

rbtChuaHoanThanh.Checked = trangThai == 0; // Kiểm tra trạng thái (0 là Chưa hoàn thành)

dTpBatDau.Value = ngayBatDau;

dTpKetThuc.Value = ngayKetThuc;

// Bạn có thể thêm xử lý khác nếu cần, ví dụ: hiển thị thêm thông tin từ các cột khác

}

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtMaCongViec.Text) ||

string.IsNullOrWhiteSpace(txtTenCongViec.Text) ||

string.IsNullOrWhiteSpace(tb\_MoTa.Text) ||

string.IsNullOrWhiteSpace(CbUuTien.SelectedItem?.ToString()) ||

dTpBatDau.Value == null ||

dTpKetThuc.Value == null)

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin trước khi reset.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string maCongViec = txtMaCongViec.Text.Trim();

string tenCongViec = txtTenCongViec.Text.Trim();

string moTa = tb\_MoTa.Text.Trim();

int uuTien = CbUuTien.SelectedIndex; // 0, 1, 2, 3

DateTime ngayBatDau = dTpBatDau.Value;

DateTime ngayKetThuc = dTpKetThuc.Value;

int trangThaiCV = rbtHoanThanh.Checked ? 1 : 0; // Cải thiện việc gán giá trị cho trạng thái công việc

// Mở kết nối và gọi stored procedure

using (SqlConnection conn = new SqlConnection(str))

{

try

{

conn.Open();

using (SqlCommand cmd = new SqlCommand("sp\_ThemCongViec", conn))

{

cmd.CommandType = CommandType.StoredProcedure;

// Truyền tham số cho stored procedure

cmd.Parameters.AddWithValue("@TenCongViec", tenCongViec);

cmd.Parameters.AddWithValue("@MoTa", moTa);

cmd.Parameters.AddWithValue("@ƯuTien", uuTien); // Truyền đúng kiểu số nguyên

cmd.Parameters.AddWithValue("@NgayBatDau", ngayBatDau);

cmd.Parameters.AddWithValue("@NgayKetThuc", ngayKetThuc);

// Thêm tham số OUTPUT cho @ref

SqlParameter refParam = new SqlParameter("@ref", SqlDbType.Bit)

{

Direction = ParameterDirection.Output // Chỉ định tham số này là OUTPUT

};

cmd.Parameters.Add(refParam);

// Thực thi stored procedure

cmd.ExecuteNonQuery();

// Lấy giá trị của tham số OUTPUT

bool refValue = (bool)refParam.Value;

if (refValue)

{

MessageBox.Show("Thêm công việc thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Tải lại dữ liệu sau khi thêm thành công

ResetForm();

}

else

{

MessageBox.Show("Lỗi khi thêm công việc. Vui lòng kiểm tra lại dữ liệu.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi thêm công việc: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

string maCongViec = txtMaCongViec.Text.Trim();

// Kiểm tra nếu mã công việc rỗng

if (string.IsNullOrEmpty(maCongViec))

{

MessageBox.Show("Vui lòng nhập mã công việc để xóa.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Câu lệnh xác nhận xóa

DialogResult dialogResult = MessageBox.Show("Bạn có chắc chắn muốn xóa công việc này?", "Xác nhận xóa", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialogResult == DialogResult.No)

{

return; // Nếu người dùng chọn "No", thoát khỏi phương thức

}

using (SqlConnection con = new SqlConnection(str)) // Kết nối tới cơ sở dữ liệu

{

try

{

con.Open();

// Câu lệnh SQL xóa công việc theo mã công việc

string query = "DELETE FROM CongViec WHERE MaCongViec = @MaCongViec";

// Tạo SqlCommand

SqlCommand command = new SqlCommand(query, con);

// Thêm tham số vào câu lệnh SQL

command.Parameters.AddWithValue("@MaCongViec", maCongViec);

// Thực thi câu lệnh xóa

int rowsAffected = command.ExecuteNonQuery();

// Kiểm tra xem có xóa được công việc không

if (rowsAffected > 0)

{

MessageBox.Show("Công việc đã được xóa thành công.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

ResetForm();

loaddata();

}

else

{

MessageBox.Show("Không tìm thấy công việc với mã đã nhập.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (SqlException ex)

{

// Xử lý lỗi SQL

MessageBox.Show($"Lỗi SQL: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

// Xử lý lỗi chung

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void button4\_Click\_1(object sender, EventArgs e)

{

string maCongViec = txtMaCongViec.Text.Trim();

string tenCongViec = txtTenCongViec.Text.Trim();

string moTa = tb\_MoTa.Text.Trim();

int uuTien = CbUuTien.SelectedIndex; // 0, 1, 2, 3

DateTime ngayBatDau = dTpBatDau.Value;

DateTime ngayKetThuc = dTpKetThuc.Value;

int trangThaiCV = 0;

if (rbtHoanThanh.Checked == true)

{

trangThaiCV = 1;

}

using (SqlConnection con = new SqlConnection(str))

{

try

{

con.Open();

// Tạo câu lệnh SQL với điều kiện LIKE cho tất cả các trường

string query = @"

SELECT MaCongViec, TenCongViec, MoTa, ƯuTien, TrangThaiCV, NgayBatDau, NgayKetThuc

FROM CongViec

WHERE (@MaCongViec IS NULL OR MaCongViec LIKE @MaCongViec)

AND (@TenCongViec IS NULL OR TenCongViec LIKE @TenCongViec)

AND (@MoTa IS NULL OR MoTa LIKE @MoTa)

AND (@UuTien IS NULL OR ƯuTien LIKE @UuTien)

AND (@TrangThai IS NULL OR TrangThaiCV LIKE @TrangThai)";

// Tạo SqlCommand với câu lệnh SQL

SqlCommand command = new SqlCommand(query, con);

// Thêm các tham số vào câu lệnh SQL

command.Parameters.AddWithValue("@MaCongViec", string.IsNullOrEmpty(maCongViec) ? (object)DBNull.Value : "%" + maCongViec + "%");

command.Parameters.AddWithValue("@TenCongViec", string.IsNullOrEmpty(tenCongViec) ? (object)DBNull.Value : "%" + tenCongViec + "%");

command.Parameters.AddWithValue("@MoTa", string.IsNullOrEmpty(moTa) ? (object)DBNull.Value : "%" + moTa + "%");

command.Parameters.AddWithValue("@UuTien", uuTien >= 0 ? (object)uuTien : DBNull.Value);

command.Parameters.AddWithValue("@TrangThai", trangThaiCV >= 0 ? (object)trangThaiCV : DBNull.Value);

// Tạo SqlDataAdapter để lấy dữ liệu

SqlDataAdapter adapter = new SqlDataAdapter(command);

// Tạo DataTable để chứa dữ liệu

DataTable dt = new DataTable();

// Đổ dữ liệu vào DataTable

adapter.Fill(dt);

// Kiểm tra nếu có kết quả

if (dt.Rows.Count > 0)

{

// Gán dữ liệu vào DataGridView

dgv.DataSource = dt;

}

else

{

MessageBox.Show("Không tìm thấy công việc nào khớp với từ khóa tìm kiếm!", "Thông báo");

}

}

catch (SqlException ex)

{

MessageBox.Show($"Lỗi SQL: {ex.Message}", "Lỗi");

}

catch (Exception ex)

{

MessageBox.Show($"Lỗi: {ex.Message}", "Lỗi");

}

}

}

// Hàm kiểm tra các ký tự SQL Injection

private bool ContainsSQLInjectionKeywords(string input)

{

if (string.IsNullOrWhiteSpace(input))

return false;

string[] keywords = { "SELECT", "INSERT", "UPDATE", "DELETE", "DROP", "ALTER", "OR", "AND", "--", ";", "/\*", "\*/" };

foreach (string keyword in keywords)

{

if (input.IndexOf(keyword, StringComparison.OrdinalIgnoreCase) >= 0)

{

return true; // Nếu tìm thấy từ khóa, trả về True

}

}

return false; // Nếu không tìm thấy từ khóa, trả về False

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

// 0. Vì MaCongViec đã set ReadOnly nên chỉ sửa các thuộc tính khác

// 1. Kiểm tra dữ liệu đầu vào

if (string.IsNullOrWhiteSpace(txtMaCongViec.Text) ||

string.IsNullOrWhiteSpace(txtTenCongViec.Text) ||

string.IsNullOrWhiteSpace(tb\_MoTa.Text) ||

string.IsNullOrWhiteSpace(CbUuTien.SelectedItem?.ToString()) ||

dTpBatDau.Value == null ||

dTpKetThuc.Value == null)

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin trước khi reset.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

string maCongViec = txtMaCongViec.Text.Trim();

string tenCongViec = txtTenCongViec.Text.Trim();

string moTa = tb\_MoTa.Text.Trim();

int uuTien = CbUuTien.SelectedIndex; // 0, 1, 2, 3

DateTime ngayBatDau = dTpBatDau.Value;

DateTime ngayKetThuc = dTpKetThuc.Value;

int trangThaiCV = rbtHoanThanh.Checked ? 1 : 0; // Cải thiện việc gán giá trị cho trạng thái công việc

// Kiểm tra dữ liệu

if (string.IsNullOrEmpty(maCongViec) || string.IsNullOrEmpty(tenCongViec))

{

MessageBox.Show("Mã công việc và tên công việc không được để trống!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (ngayKetThuc < ngayBatDau)

{

MessageBox.Show("Ngày kết thúc không được trước ngày bắt đầu!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

// Câu lệnh SQL Update

string query = @"

UPDATE CongViec

SET TenCongViec = @TenCongViec,

MoTa = @MoTa,

ƯuTien = @UuTien,

NgayBatDau = @NgayBatDau,

NgayKetThuc = @NgayKetThuc,

TrangThaiCV = @TrangThaiCV

WHERE MaCongViec = @MaCongViec";

using (SqlConnection conn = new SqlConnection(str))

{

conn.Open();

using (SqlCommand cmd = new SqlCommand(query, conn))

{

// Gán tham số

cmd.Parameters.AddWithValue("@MaCongViec", maCongViec);

cmd.Parameters.AddWithValue("@TenCongViec", tenCongViec);

cmd.Parameters.AddWithValue("@MoTa", moTa);

cmd.Parameters.AddWithValue("@UuTien", uuTien);

cmd.Parameters.AddWithValue("@NgayBatDau", ngayBatDau);

cmd.Parameters.AddWithValue("@NgayKetThuc", ngayKetThuc);

cmd.Parameters.AddWithValue("@TrangThaiCV", trangThaiCV);

// Thực thi câu lệnh

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Cập nhật thành công!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loaddata(); // Nếu cần cập nhật lại dữ liệu sau khi chỉnh sửa

ResetForm(); // Nếu cần reset form sau khi hoàn tất cập nhật

}

else

{

MessageBox.Show("Không tìm thấy công việc cần cập nhật!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

}

catch (SqlException sqlEx)

{

// Xử lý lỗi SQL

MessageBox.Show($"Lỗi SQL: {sqlEx.Message}", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

// Xử lý lỗi chung

MessageBox.Show($"Lỗi: {ex.Message}", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

}

private void textBox\_TrangThaiCV\_TextChanged(object sender, EventArgs e)

{

}

private void dateTime\_NgayKetThuc\_ValueChanged(object sender, EventArgs e)

{

}

private void label\_ƯuTien\_Click(object sender, EventArgs e)

{

}

private void label\_MaCongViec1\_Click(object sender, EventArgs e)

{

}

private void tb\_MaCongViec1\_TextChanged(object sender, EventArgs e)

{

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button1\_Click\_2(object sender, EventArgs e)

{

}

private void tb\_MaNhanVien1\_TextChanged(object sender, EventArgs e)

{

}

/\*private void button\_Khoitao1\_Click(object sender, EventArgs e)

{

tb\_MaCongViec2.Text = string.Empty;

tb\_TenCongViec2.Text = string.Empty;

tb\_MoTa.Text = string.Empty;

tb\_Ưutien.Text = string.Empty;

dateTime\_NgayBatDau.Value = DateTime.Now;

dateTime\_NgayKetThuc.Value = DateTime.Now;

tb\_trangthai.Text = string.Empty;

tb\_MaNhanVien1.Text = string.Empty;

}\*/

private void tb\_Ưutien\_TextChanged(object sender, EventArgs e)

{

}

private void ResetForm()

{

// Xóa nội dung của TextBox

txtMaCongViec.Text = string.Empty;

txtTenCongViec.Text = string.Empty;

tb\_MoTa.Text = string.Empty;

// Đặt ComboBox về giá trị mặc định

if (CbUuTien.Items.Count > 0)

{

CbUuTien.SelectedIndex = 0; // Giá trị đầu tiên trong ComboBox

}

// Đặt lại DateTimePicker về ngày hiện tại

dTpBatDau.Value = DateTime.Now;

dTpKetThuc.Value = DateTime.Now;

// Đặt RadioButton về trạng thái mặc định

rbtChuaHoanThanh.Checked = true; // Mặc định là "Chưa hoàn thành"

rbtHoanThanh.Checked = false;

}

private void Reset\_click(object sender, EventArgs e)

{

loaddata();

ResetForm();

}

private void grbox\_Côngviec\_Enter(object sender, EventArgs e)

{

}

private int ConvertUuTienTextToNumber(string CbUuTien)

{

switch (CbUuTien)

{

case "Nghiêm trọng":

return 0;

case "Cao":

return 1;

case "Trung bình":

return 2;

case "Thấp":

return 3;

default:

return -1; // Hoặc giá trị mặc định nếu không nhận diện được

}

}

private string ConvertUuTienNumberToText(int uuTien)

{

switch (uuTien)

{

case 0:

return "Nghiêm trọng";

case 1:

return "Cao";

case 2:

return "Trung bình";

case 3:

return "Thấp";

default:

return "Không xác định"; // Trường hợp không phải 0, 1, 2, 3

}

}

}

}